

Energiahatékony útválasztás IoT környezetben

Energiahatékony útválasztás

Pásztor Dániel

BME-AUT

Doktorandusz

pasztor.daniel@aut.bme.hu

1117 Bp.

Magyar tudósok krt. 2 QB231

(levelező szerző)

Dr. Ekler Péter

BME-AUT

Egyetemi docens

ekler.peter@aut.bme.hu

1117 Bp.

Magyar tudósok krt. 2 QB226

Dr. Levendovszky János

BME-HIT

Tudományos és innovációs

rektorhelyettes

levendov@hit.bme.hu

1117 Bp.

Magyar tudósok krt. 2 IB129

Absztrakt

Az Ipar4.0 egyik alapvető célja a termelési folyamatok összehangolása, a különböző gyártási eszközök automatizálása és felügyelete. Ennek elengedhetetlen feltétele a szenzorok telepítése, illetve a mért adatok hatékony begyűjtése és feldolgozása.

A szenzoradatok beolvasására és továbbküldésére hatékony megoldást kínálnak az IoT eszközök. Bizonyos alkalmazásokban az eszközök energiaellátása csak saját telep által biztosított, amely az energiahatékony kommunikációs protokollok alkalmazását igényli.

Kutatásaink során olyan új algoritmust fejlesztettünk vezeték nélküli IoT eszközök kommunikációjára, amely minimális energia felhasználással biztosít megbízható információátvitelt. Pontosabban, a szenzoriális információt csomagokban továbbítva, a továbbításhoz szükséges energia minimalizálása mellett garantáljuk, hogy a csomagokban lévő információ előírt valószínűséggel jut el a bázisállomásba. Ezzel a hálózat élettartamát és információátviteli képességét sikerül maximalizálni, hiszen az alacsony energia fogyasztással az IoT eszközök adattovábbítási képessége jelentősen meghosszabbítható.

Címszavak: IoT, szenzorhálózat, energiahatékony, útválasztás, WiFi

Abstract

One of the key aspects of the fourth industrial revolution (Industry 4.0) is the automated data acquisition and extensive data analysis for monitoring complex manufacturing processes. Hence, efficient data collection is the core concept of implementing Industry4.0 on IoT platforms. This requires energy aware communication protocols for Wireless Sensor Networks (WSNs) where the sensing and processing on the IOT node is supplied only by local battery power.

In our research, we have developed novel routing algorithms which guarantee minimum energy data transfer subject to pre-defined reliability constraints. Data is transmitted in the form of packets and the routing algorithm identifies the paths over which the packets can reach the Base Station (BS) with minimum transmission energy, while the probability of successful packet transmission exceeds a pre-defined reliability parameter. In this way, the longevity and the information throughput of the network is maximized, as low energy transmission will considerably extend the lifetime of the IoT nodes.

Title: Energy-efficient routing in Wireless Sensor Networks

Keywords: IoT, WSN, energy-efficient, routing, WiFi

Bevezetés

Az Ipar4.0 egyik fő pontja a termelési folyamatok összehangolása, a különböző gyártási eszközök automatizálása és felügyelete. Ennek elengedhetetlen feltétele a különböző szenzorok telepítése, illetve a mért adatok begyűjtése és feldolgozása.

Sok esetben óriási költségekkel járna minden egyes szenzor bekötése a vezetékes hálózatba, ha egyáltalán fizikálisan megvannak rá a lehetőségek. Ilyenkor hatékony megoldást kínálnak az IoT eszközök, melyek képesek kezelni az adott szenzort, majd az innen származó értékeket vezeték nélküli hálózaton tudják továbbítani. Ez egy hatalmas rugalmasságot is jelent a rendszernek, gyakorlatilag szükség szerint lehet a hálózatot további szenzorokkal bővíteni.

Sajnos a vezeték nélküli eszközöknek megvan a maguk hátránya is: valamilyen módon szükséges az energiaellátásuk biztosítása, mely jellemzően egy beépített akkumulátor segítségével történik. Ezeket az energiaforrásokat karban kell tartani, például a kapacitás függvényében bizonyos időközönként újra kell tölteni őket. Ilyen körülmények mellett mindenképp célpont az eszközök energiahatékony működése.

Az eszközök energiafelhasználása két fő kategóriára bontható: a szenzor működtetéséhez, illetve az adatok továbbításához szükséges energia. A szenzor működtetése több külső paramétertől függ, mint például a választott alkatrészek fogyasztása, a szenzor működési módja, illetve a vezérlő elektronika különböző paraméterei. Ezzel szemben a kommunikáció során szükséges energia jól definiálható, mely jellemzően a távolság, illetve a környezetbeli elhelyezkedéstől függ.

Kommunikáció során sokszor előnytelen lehet, ha az adott szenzor közvetlenül a központnak küldi az üzenetet. Ekkor a távolabb elhelyezett szenzoroknak nagyobb energiával kell küldeniük a csomagukat, így hamarabb is merülnek le. Helyette érdemes lehet megvizsgálni azt az esetet, amikor az egyes szenzorok kooperatívan, vagyis egymás segítve juttatják el az adott csomagot a központnak, így jobban eloszlatva az energiafelhasználást.

Kapcsolódó munkák

A nagy méretű szenzorhálózatok energiahatékony üzenettovábbításra több megoldást is kidolgoztak különböző szempontok figyelembevételével.

A LEACH [1] protokoll fordulókra bontja az adattovábbítást. Minden fordulóban a csomópont egy sztochasztikus algoritmus segítségével eldönti magáról, hogy továbbító bázis lesz-e. Ezek után minden egyéb csomópont a hozzá legközelebb lévő továbbító bázisnak küldi a saját csomagját, időben ütemezve. A továbbító bázisok az összegyűjtött csomagokhoz hozzáteszik a saját csomagjukat, végrehajtanak az egész adathalmazon egy tömörítési algoritmust, majd az így kapott tömörített csomagot továbbítják a központ felé.

A *LEACH* algoritmusnak számos előnnyel rendelkezik a hagyományos közvetlen üzenetküldéssel szemben. A kiválasztó algoritmus hangolásával állítható, hogy egy fordulóban várhatóan hány továbbító bázis legyen kiválasztva. Ennek köszönhetően ezek a csomópontok kis energiaigénnyel tudják továbbítani a csomagjukat a továbbküldő bázis felé. Az összegyűjtött csomagok tömörítése pedig a továbbküldést végző adók energiaigényét csökkenti.

Az algoritmus hátránya közé tartozik viszont, hogy nem képes a csomópontok energiaszintjét figyelembe venni. A kiválasztás során előfordulhat, hogy egy már alacsony energiaszinten lévő adó kerül kiválasztásra, mely nem lesz képes továbbítani az összegyűjtött csomagokat, így értékes információ veszhet el a rendszerből. Emellett a véletlenszerű kiválasztó algoritmus nem garantálja, hogy a továbbító bázisok egyenletesen legyenek elosztva a szenzorhálózatban, így egy fordulón belül létre jöhetnek gócpontok, melyek a hálózat hatékonyságát rontják.

A *PEGASIS* [2] algoritmus egy teljesen más megközelítést alkalmaz az adatok továbbítására. Itt a fő hangsúly nem az egyes szenzorokon található, hanem a belőlük kiszámítható aggregált adatokon, mint például egy mért érték minimuma, maximuma vagy átlaga. Hasonlóan a *LEACH* algoritmushoz, itt is fordulókról van szó, melyben minden csomópont a mért értékét juttatná el a központhoz.

A *PEGASIS* törekszik arra, hogy egy fordulón belül csak egy csomópontnak kelljen nagyobb energiával üzenetet küldeni. Ehhez egy mohó algoritmus segítségével egy láncot konstruál, melyben minden csomópont megtalálható, és törekszik arra, hogy a csomópontok közötti távolság minimális legyen. Minden forduló elején a soron következő csomópont ki lesz jelölve főadónak. A lánc két végéről indulva minden csomópont kombinálja a beérkezett adatot a saját mért szenzoradataival, majd ezt továbbítja, így az adatcsomagok mérete állandó marad. Miután a főadó megkapta mindkét irányból az aggregált adatokat, összegzi őket, és továbbküldi a központ felé.

A *LEACH* algoritmushoz hasonlóan jelentős előnyt biztosít a *PEGASIS* az aggregált adatok összegyűjtésekor, ha a központi egységnek ezekre van szüksége. Mivel a kommunikáció során már csak az aggregált adatok kerülnek továbbításra, illetve a kommunikáció során általában egymáshoz közel lévő csomópontok kommunikálnak, ezért energiahatékony megoldást biztosít az algoritmus. Arra az esetre viszont, amikor szükségünk van minden egyes csomópont beolvasott értékére, nem használható.

Modell

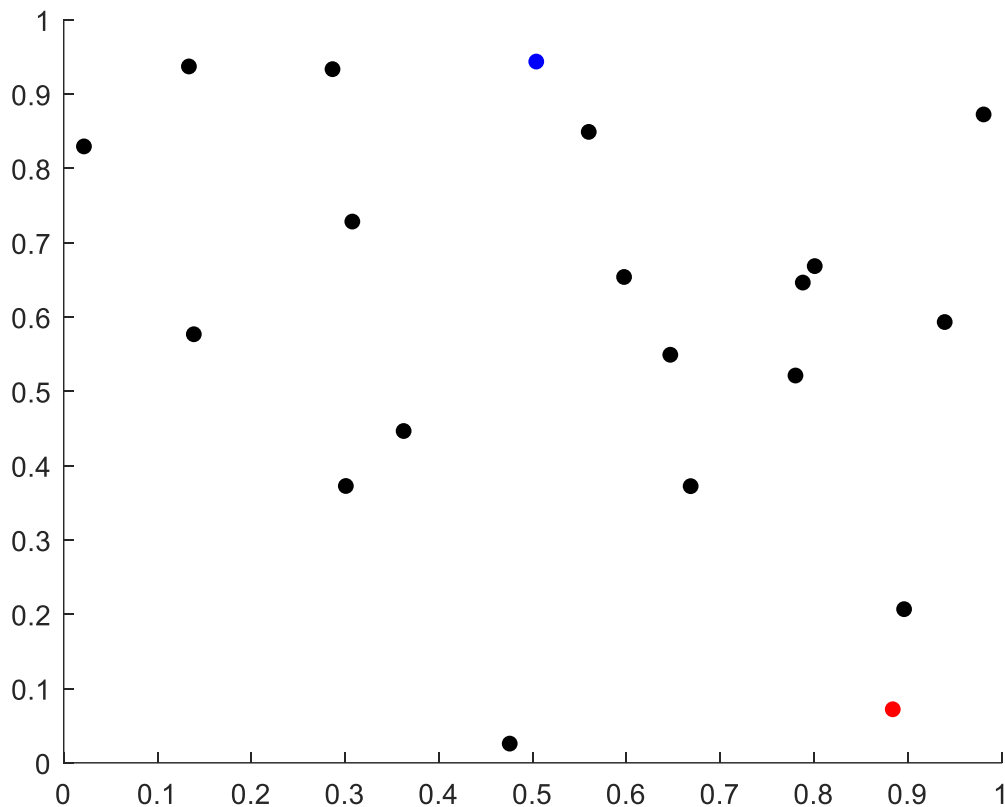
Az ismertetett algoritmusok azzal a feltételezéssel dolgoztak, hogy két csomópont között a távolsággal arányos energia kell a csomag biztos átküldéséhez, viszont nem veszi figyelembe a csomag elvesztésének valószínűségét. Az általunk felhasznált modellben feltételezzük, hogy az adók általuk választott energia felhasználásával küldhetnek csomagokat, így megválaszthatják a sikeres

kommunikáció valószínűségét. A valószínűség pontos megállapításához a Rayleigh-féle jelgyengülési modellt használjuk fel:

$$g_{ij} = -d_{ij}^{\alpha} \frac{Q S_z^2}{\ln(P_{ij})},$$

ahol g jelöli a felhasznált energiát, d a távolságot, α a dimenziószámot, Θ és σ a terjedési modell paraméterei, P pedig a sikeres kommunikáció az i és j csomópontok között.

A szenzorhálózatunk álljon N egységből, illetve a központból, mely begyűjti a szenzorok által mért értékeket. Jelenleg minden adót véletlenszerűen helyezünk el egy egységnyi méretű négyzet területén. Minden állomás egy megadott E kezdőenergiával rendelkezik, majd véletlenszerű sorrendben üzenetet küldenek a központ felé úgy, hogy egy előre megadott P_s valószínűséggel sikeresen megérkezzen. Kihasnálva, hogy a csomópontok egymás között is kommunikálhatnak, addig vizsgáljuk az elküldött üzenetek számát, ameddig a soron következő adó már nem képes a rögzített valószínűséggel üzenetet küldeni.



1. ábra: Példa az adók elhelyezkedésére, pirossal jelölve a központ, késsel pedig a következő üzenet küldőjét

A kidolgozott algoritmusaink arra törekednek, hogy az üzenetek lehetőleg az aktuálisan nagyobb energiával rendelkező csomópontokon keresztül történjen, így lehetőleg a teljes hálózatra

nézve az egyes energiaszintek együtt csökkennek, nem lesznek nagy különbségek. Ennek érdekében a felhasznált energia minimalizálása helyett a kommunikáció után fennálló legkisebb energiaszint maximalizálására törekszünk. Belátható, hogy tetszőleges számú csomóponton keresztüli kommunikáció során a keletkező energiaszintek közül a legkisebb akkor lesz maximális, ha a részt vevő összes csomópont ugyanarra az energiaszintre kerül.

A bizonyítása indirekt módon történik: tegyük fel, hogy a kommunikáció után keletkező energiaszintek nem egyenlőek, de a legalacsonyabb energiaszint így lesz a maximális. Ez esetben biztos lesz legalább egy adó, mely nagyobb energiaszinttel rendelkezik, mint a minimális (különben egyforma energiaszinten lenne az összes adó). Ha ez az adó nagyobb energiával továbbítaná a csomagot (de még így sem érné el a minimumszintet), a minimumszinten lévő adóknak elég kisebb energiával részt venniük a kommunikációban ugyanakkora sikeres küldési valószínűség mellett, így tehát növeltük a minimum energiaszintet, mely ellentmondásban áll az eredeti feltevessel.

Javasolt algoritmusok

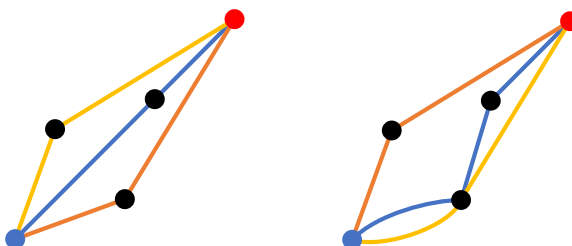
Az algoritmusaink ezek alapján arra törekednek, hogy egy üzenetküldés során a kommunikációban részt vevő adók energiaszintje a küldés után egy közös energiaszintre kerüljenek, és azok a csomópontok fognak részt venni, melyekkel az előálló közös energiaszint a legnagyobb. Bár előnyös, ha egy kommunikáció során több adó is segít az üzenet továbbításában, érdemes lehet az adók számának maximalizálása, mivel bár több adó segítségével magasabb minimum energiaszintet leszünk képesek elérni, az egész rendszerre vonatkoztatva több energia felhasználásával is jár. Ennek megfelelően a maximális csomópontok száma alapján különböző algoritmusokról beszélhetünk:

Direkt küldés esetén a forrás csomópont direkt a bázis állomásnak küldi az üzenetet, köztes csomópont használata nélkül. Ez tekinthető az alap stratégiának, aminél jobb eredményeket szeretnénk elérni.

2-ugrásos (2-hop) algoritmusnál a küldés legfeljebb egy köztes csomóponton keresztül történik. Az optimális küldéshez az összes csomópontra egyenként kiszámoljuk a küldés után kialakuló megmaradó közös energiaszintet, melyhez egy másodfokú polinom egyenletet kell megoldani. Az így kapott energiaszintek közül a legmagasabb eredménnyel rendelkező csomópontot választjuk köztes csomópontnak.

Sokugrásos (multi-hop) algoritmus esetén tetszőleges számú köztes csomópont lehet tetszőleges sorrendben, majd itt is a legmagasabb megmaradó közös energiaszinttel rendelkező konfigurációt választjuk. Ennek a pontos megoldása rengeteg számítással járna, többek között rengeteg n -ed fokú teljes polinom megoldását igényelné, ezért a feladatot az általunk választott energiaszint mellett egy additív metrikán alapuló útvonalkeresésébe képezzük le, amely tradicionális

algoritmusokkal (pl. Bellman-Ford, vagy Dijkstra) a node-ok számában polinomiális idő alatt adja meg a közelítőleg optimális útvonalat.



2. ábra: 2-ugrás és multi-ugrás példa. Egy szín egy csomagnak az újtát mutatja.

Az ismert algoritmusokat a MATLAB környezetben megvalósítottuk, majd ebben végeztünk szimulációkat. Több mérést végeztünk az algoritmusokkal különböző adóelrendezések mellett, illetve különböző üzenetküldési sorrend mellett, és vizsgáltuk az üzenetküldések számát az első adó lemerüléséig. Az ezekből az adatokból készített statisztikák láthatóak az 1. táblázatban.

1. táblázat: Ugrásszám statisztika különböző adószám és algoritmusok esetén

	Minimum			Átlag			Maximum		
<i>Csp.</i>	<i>Direkt</i>	<i>2-ugr</i>	<i>Multi</i>	<i>Direkt</i>	<i>2-ugr</i>	<i>Multi</i>	<i>Direkt</i>	<i>2-ugr</i>	<i>Multi</i>
10	26	76	75	112.2	160.32	152.09	289	416	398
20	78	220	136	212.39	350.79	281.53	536	761	608
50	156	590	314	482.7	987.3	656.3	1392	1896	1443
100	308	1256	650	760.2	1980.2	1178.9	1673	3622	2287

Látható, hogy a felsorolt algoritmusok közül mindegyik statisztikát figyelembe véve a *2-ugrás* algoritmus a legjobb választás. *Direkt* küldés esetén nagyon hamar lemerülnek a távolabbi csomópontok, melyeknek a távolság miatt nagyobb energiabefektetéssel kell az üzeneteket küldeniük.

A *multi-ugrás* algoritmus bár jobban teljesít a *direkt* algoritmusnál, de az adók számának növekedésével egyre inkább elmarad a *2-ugrás* algoritmustól. Ez a megfigyeléssel magyarázható, hogy ha egy alacsony energiaszintű adó szeretne egy üzenetet küldeni, akkor az algoritmus tulajdonsága miatt a kommunikációban résztvevő összes csomópont energiaszintjét a le fogja húzni a saját energiaszintje alá. *2-ugrás* esetén ez csak egy köztes csomópontot érint, míg *multi-ugrás* esetén sokkal több, szélsőséges helyzetben akár az összes csomópontra hatással van.

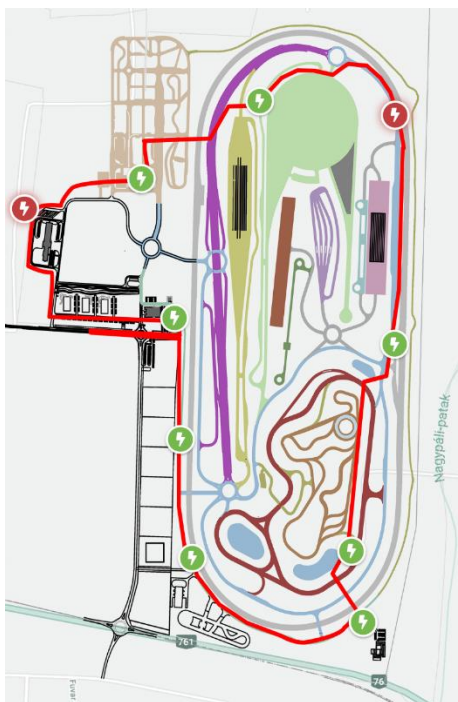
Jövőbeli lehetőségek

Mint megállapítottuk, a felsorolt algoritmusok közül a *2-ugrás* tekinthető a legjobb választásnak minden vizsgált körülmény között. A jövőben szeretnénk vizsgálni, hogyan változnak ezek az eredmények, ha a kettő helyett három, négy, vagy még több, de fix számú ugrást engedünk, melyet *k-ugrás* (*k-hop*) algoritmusnak neveztünk el. Sejtésünk szerint az elhelyezett adók sűrűségének

növekedésével valamilyen arányban fog az optimális k érték is növekedni, melyet mindenképp igazolni szeretnénk.

Ezek mellett az ismertetett eredmények véletlenszerűen elhelyezett csomópontokkal lettek mérve. A jövőben mindenképp szeretnénk a szenzorhálózat topológiáját, illetve üzenetküldési gyakoriságot is figyelembe venni, arra vonatkozólag egy optimális üzenetküldési stratégiát ajánlani. Az elkészített modellt tovább lehet bővíteni az adók közötti akadályok bevezetésével (például épületek), ennek megfelelően javítani az algoritmusokon.

Az egyetem is közreműködik a Zalaegerszegi önvezető autók teszt pályájának kivitelezésében és fenntartásában, mely egyelőre egyedülálló Európában. A pályán rengeteg szenzor található, melyek mérései elengedhetetlen részét képezik a tesztelő csapatok elemzéseinek. Terveink szerint itt is szeretnénk felhasználni a kutatási eredményeinket.



3. ábra: A zalaegerszegi önvezető jármű tesztpálya térképe

Köszönetnyilvánítás

A kutatás a BME-Mesterséges Intelligencia FIKP EMMI (BME FIKP-MI/SC), valamint a Magyar Tudományos Akadémia Bolyai János Kutatói Ösztöndíjának támogatásával készült.

Irodalomjegyzék

- [1] A. C. & H. B. W. R. HEINZELMAN, „Energy-efficient routing protocols for wireless microsensor networks,” in *33rd Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, 2017.

- [2] S. & R. C. S. LINDSEY, „PEGASIS: Power-efficient gathering in sensor information systems,” in *IEEE Aerospace Conference*, Big Sky, MT, USA, USA, 2002.